NATIONAL SECURITY AGENCY
CENTRAL SECURITY SERVICE
FORT GEORGE G. MEADE, MARYLAND 20755-6000

FOIA Case: 19136E
24 September 2007

Mr. John L. Young
Cryptome
Suite 6E
251 West 89th Street
New York, NY 10024

Dear Mr. Young:

This is our final response to your Freedom of Information Act (FOIA) request of 16 November 2000, which you narrowed to "documents in the year 1970 concerning 'non-secret encryption' described in the paper by J.H. Ellis, 'The Possibility of Secure Non-Secret Digital Encryption,' CESG Report, January 1970, and documents concerning public key cryptography by the USA or other nations. Should NSA have no documents about 'non-secret digital encryption' and public key cryptography for the year 1970," then you requested "documents on the topics from the earliest year other than 1970." A copy of this narrowed request is enclosed.

As you were previously advised, for purposes of this request and based on the information you provided in your letter, you are considered an "all other" requester. As such, you are allowed two hours of search and the duplication of 100 pages at no cost. Taking this into account, final costs of $810.75 assessed include $807.30 for search, and $3.45 for duplication of 23 pages in excess of 100. You have already remitted $819.30 for search and estimated duplication costs associated with this case. Because the actual duplication costs are slightly less than estimated you will be sent a refund of $8.55 under separate cover.

We provided you a response regarding most of the responsive documents on 13 February 2007. At that time, we advised you that some documents had been referred to another agency for consultation prior to final release. Those documents have been reviewed under the FOIA, and four documents are enclosed. Certain information, however, has been deleted from the enclosures, and one document (18 pages) has been withheld in its entirety.

Some of the information deleted from the enclosures, as well as the fully withheld document, was found to be currently and properly classified in accordance with Executive Order 12958, as amended. This information meets the criteria for classification as set forth in Subparagraphs (c) and (g) of Section 1.4 and remains classified TOP SECRET and CONFIDENTIAL as provided in Section 1.2 of the Executive Order. The information is classified because its disclosure could reasonably be expected to cause exceptionally grave damage to the national security. Because the information is currently and properly classified, it is exempt from disclosure pursuant to the first exemption of the FOIA (5 U.S.C. Section 552(b)(1)).

In addition, this Agency is authorized by various statutes to protect certain information concerning its activities. We have determined that such information exists in all five documents. Accordingly, those portions are exempt from disclosure pursuant to the third exemption of the FOIA which provides for the withholding of information specifically protected from disclosure by statute. The specific statutes applicable in this case are Title 18 U.S. Code 798; Title 50 U.S. Code 403-1(i); and Section 6, Public Law 86-36 (50 U.S. Code 402 note).

Information has also been protected from the document that is denied in its entirety pursuant to the fifth exemption of the FOIA. This exemption applies to inter-agency or intra-agency memoranda or letters which would not be available by law to a party in litigation with the agency, protecting information that is normally privileged in the civil discovery context, such as information that is part of a predecisional deliberative process.

Since one document was withheld in its entirety and information was withheld from the enclosures, you may construe this as a partial denial of your request. You are hereby advised of this Agency's appeal procedures. Any person denied access to information may file an appeal to the NSA/CSS Freedom of Information Act Appeal Authority. The appeal must be postmarked no later than 60 calendar days from the date of the initial denial letter. The appeal shall be in writing addressed to the NSA/CSS FOIA Appeal Authority (DJ4), National Security Agency, 9800 Savage Road STE 6248, Fort George G. Meade, MD 20755-6248. The appeal shall reference the initial denial of access and shall contain, in sufficient detail and particularity, the grounds upon which the requester believes release of the information is required. The NSA/CSS Appeal Authority will endeavor to respond to the appeal within 20 working days after receipt, absent any unusual circumstances.

The Federal Bureau of Investigation (FBI) has asked that we protect information pursuant to 5 U.S.C. 552 (b)(1) from the enclosures. Those deletions have been marked with the code OGA (Other Government Agency). Any appeal of the denial of FBI information should be directed to the Director, Office of Information and Privacy, U.S. Department of Justice, 1425 New York Ave., NW, Suite 11050, Washington, DC 20530-0001, within 60 days from the receipt of this letter. The envelope and the letter should be clearly marked "Freedom of Information Appeal" or "Information Appeal" and the FBI's FOIPA number 1070885-000 should be cited.

Sincerely,

Rhea D. Siers

RHEA D. SIERS
Deputy Associate Director for Policy

Encls:
a/s

CRYPTOME 251 WEST 89TH ST SUITE 6E NEW YORK NY 10024 212-873-8700

Attention: FOIA Office

September 8, 2001

By fax to: 301-688-4762

Pamela N. Phillips
Chief, FOIA/PA Services
FOIA Office (DC321)
National Security Agency
9800 Savage Road STE 6248
Ft. George G. Meade, MD 20755-6248

Re FOIA Case: 19136

Dear Ms. Phillips,

In response to my telephone conversation yesterday with Vivian from your office I further narrow my request to most quickly obtain information requested by my letter of November 16, 2000:

> The invention, discovery and development of "non-secret encryption" (NSE) and public key cryptography (PKC) by United Kingdom, United States, or any other nation's intelligence and cryptology agencies, prior to, parallel with, or subsequent to, the PKC work of Diffie-Hellman-Merkle.

1. I request documents in the year 1970 concerning "non-secret digital encryption" described in the paper by J H Ellis, "The Possibility of Secure Non-Secret Digital Encryption," CESG Report, January 1970, and documents concerning public key cryptography by the USA or other nations.

2. Should NSA have no documents about "non-secret digital encryption" and public key cryptography for the year 1970, then I request documents on the topics from the earliest year other than 1970.

Thank you very much.

Sincerely,

John Young
E-mail: jya@pipeline.com
Fax: 212-787-6102

~~TOP SECRET//MR~~

(b)(1)
(b)(3)-P.L. 86-36

Proof of Concept Issues

Internal Issues

[redacted]

External Issues

- Modular PKI meets most requirements of Law Enforcement.
- Nullifies industry objections of cost and viability.
- Modularity puts key escrow debate back into policy and takes it out of technology.
- Design uses accepted standards and protocols.
- PKI allows for arbitrary encryption algorithms.
- Only design for which an individual's secrets *never* come together in key recovery process -- this makes abuse of system much harder to accomplish.
- Data recovery and key recovery are accomplished through distinct mechanisms.
- Can be supported via hardware or software.
- Proof of concept demo marries in house development with commericially available products: a prototypical example of the best way to proceed.

ACTIONS:

- Prototype Modular PKI for internal security solution. Software first, hardware to follow. Q involvement essential. Make hard decision on FORTEZZA solution.
- Take PKI demo to government agencies to demonstrate ease of use and adherence to public key recovery guidelines and get feedback.
- Establish commercial awareness of solution; work together to build products (provided they are guaranteed a market).

~~FOR OFFICIAL USE ONLY~~

~~TOP SECRET//MR~~

A Modular Public Key Infrastructure for Security Management

Proof of Concept Proposal

(b)(1)
(b)(3)-P.L. 86-36

Abstract

A six month plan is outlined for developing and demonstrating a fully functional prototype software suite based on the [redacted] *modular public key infrastructure (PKI). The prototype will support key escrow for law enforcement, a separate data recovery component for restoration of archived information, secure message encryption for electronic mail (e-mail), and strong authentication. Initially, two parallel approaches are advocated. One team will investigate the technical validity (and viability) of the design using in-house developed code leading to a prototype which will demonstrate the unique features of this PKI. The other team will determine the feasibility of using commercial off the shelf (COTS) products to implement the infrastructure. Time permitting, a fully interoperable prototype, using commercially available products, will be implemented and demonstrated.*

The current Modular PKI description is similar to an Internet Engineering Task Force (IETF) draft. It provides a general blueprint of a PKI without the hardware/software implementations that demonstrate its viability. This project will determine that viability, and if successful, be sufficient evidence to consider pushing the PKI draft to the next natural stage: an IETF Request for Comment (RFC).

Introduction

The [redacted] Modular Public Key Infrastructure for Security Management, henceforth referred to as the PKI, is an exciting development in the attempt to achieve network-wide secure, authenticated communications while simultaneously addressing the needs of law enforcement and civil libertarians. There are several striking features of this design, not the least of which is its *modularity*: the ability of users to pick and choose the features that they need (or are required to have).

The PKI design is currently patent pending. Its release to a NIST technical advisory group, meeting to determine data recovery standards for the next Federal Information Processing Standard (FIPS), is likely. Other venues are being considered for its release, including EUROCRYPT and the Public-Key Solutions Conference.

The next logical step in the evolution of this PKI is the construction of a working prototype, programmed in software, to run on various platforms such as UNIX-based SUN workstations or Windows-based PCs. In preparation a one-week effort by a small technical group (intimately involved with either the design of NSA's ICARUS e-mail system or the [redacted] PKI) was convened to develop a set of specifications for both a prototype and a demonstration of the unique features of this PKI. The charge to the group was to lay the groundwork for demonstrating proof of concept in two areas:

- Validate, through implementation, the features and technical demands of the PKI.
- Determine the extent to which the PKI can be built - and maintained - with commercial off the shelf (COTS) products.

The group determined that a concentrated six-month effort involving approximately ten people would suffice to meet the stated goals. This effort is proposed to commence in June 1997 [redacted] drawing expertise and personnel from C, Q, R, X, [redacted] groups.

The following section outlines, in considerably more detail, the proof of concept proposal. It includes a timetable and names of individuals identified as potential team members. A companion draft outlining the *technical blueprint* of the proposal is also in final draft form and is available. The latter spells out specific implementation details that the technical team will follow in producing the prototypes (for example, X.509 certificate use, S/MIME, crypto-engines, signing protocols and parameters, etc.) While choices of some components were arbitrary, others were specifically driven by what we felt were likely to be offered in a COTS environment. Likewise, despite the fact that the PKI design is modular and will support multiple components, it was decided to limit the prototype and demonstration efforts so that a fully functional product could be produced quickly. Based on the level of success achieved, future expanded efforts may be warranted.

Proof of Concept Proposal

(b)(1)
(b)(3)-P.L. 86-36

Goals

The prototype and demonstration are designed to provide a proof of concept for

- the technical viability of the PKI design to meet the stated goals, and
- the ability to use commercial off the shelf (COTS) products to implement the design.

One of the unique features of the [redacted] PKI is the separation of the mechanisms which provide data recovery for the user from those needed to service warranted law enforcement access. This is a crucial feature that the group felt must be demonstrated.

Other general design criteria for the PKI were

- All components of the PKI will be public.
- The default protocols and algorithms will have been thoroughly vetted in the public domain.
- The infrastructure must be scalable to accommodate a large number of users.
- Secure communication must not *require* the recipient to play an active part in the key exchange (i.e., a common session key can be computed by the sender alone).
- Interoperability, at the cost of increased overhead, can be achieved with *key encryption key* (KEK) systems, such as *Royal Holloway.*

The group decided that the prototype needed to demonstrate an easy-to-use infrastructure supporting the normal functions of authentication and confidentiality while highlighting the key recovery feature. *The major service this prototype will demonstrate is secure e-mail.* The escrow functions will be implemented, and time permitting, demonstrated, but this will not be an overall goal. In addition, the prototype will initially serve only a small number of users. Issues of scalability will be addressed in terms of the network services that need to be in place to accommodate larger groups.

By exploring the use of COTS products, the requirements covering use of publicly known and vetted protocols and unregulated algorithms can be examined. Demonstrating use of such products might also help answer questions of scalability. Finally, noting any failures of the commercial market to meet the development of this PKI may help drive product redesigns to our benefit.

Major program components

Based on the above, the group determined that the following modules will be needed to implement the prototype and perform the demonstration specified.

1.A Client e-mail application which will perform:
Standard e-mail functions,
Authentication and verification through digital signatures of e-mail messages, and
Confidentiality of message content through encryption.

2.A client level key-management module which will perform the following services:
Enrollment of client for signature and confidentiality services,

Message key recovery, especially to uncover a users securely stored secret keys in the event of a forgotten pass phrase, and
Certificate management, including address book management and possibly caching of certificates.

3.A server module to implement Message Key Recovery Center Functions.

4.A server module to implement Certification Authority Functions, including mechanisms for
Authenticating users,
Providing directory services - public for users to obtain certificates, and private for holding sensitive enrollment information, and
Generating and signing users certificates.

5.A server module for the Escrow Agent which can
Verify signatures as a means of authenticating a request,
Form public parameters from the users secret parameters,
Sign the generated public parameter and return the signed message to the user,
Escrow the users private information into its database, and
Develop partial session key variables when confronted with a warranted request.

Strategy

To demonstrate the validity of the PKI design, a small group of individuals will be tasked to develop an in-house, home-grown prototype, probably developed on the NSA Classified Network, and leveraging, whenever possible, readily available packages for performing the required functions and services. The ultimate goal of this team is a working prototype meeting the demonstration specifications.

To determine if COTS products are capable of being used to build the system, another small team of individuals will study several readily available packages that could possibly be used to implement the PKI. This will involve purchasing these packages and attempting to use them to build some of the various modules. The main goal here is to determine the feasibility of using COTS, or, if not, what pieces of the PKI cannot be so supported. If time permits and a set of COTS products can be found to build a prototype, an effort will be made to do so.

If both efforts succeed in developing prototypes, and, if time permits, an attempt will be made to demonstrate interoperability. Once again, ***the ultimate goal is to realize the PKI in a COTS environment.***

Schedule

In the April/May time frame, it is expected that

1.The proposal will migrate from draft to finished document,
2.Team leaders will be determined for each effort,
3.Additional effort ("homework") will be performed to further specify the prototype and demonstration details,
4.A list will be generated of hardware/software requirements needed to support the effort, and
5.A list of team members will be formed.

It is expected the actual effort will start in June and probably last from 4 to 6 months, depending on the availability of the team participants.

Human Resources

The group cited the following individuals to be contacted for staffing the effort. It is expected that only a small number of these individuals will be available on a full time basis. Team leaders for the pre-start-up phase will be [redacted] for the in-house effort and [redacted] for the COTS development team.

(b)(3)-P.L. 86-36

In-house Development		COTS Development	
[redacted]	E21	[redacted]	[redacted]
	[redacted]		R2
	Q5		R21
	[redacted]	K12	
		C43	
	Q5	[redacted]	Q6
	ISMC		X2

In addition, a small number of junior personnel, interns and summer program participants will be asked to join in the effort, both to help with the programming and as an educational and training experience. The overall effort will be mentored by [redacted] [redacted]

(b)(1)
(b)(3)-P.L. 86-36

A Modular Public Key Infrastructure for Security Management

1. Key Escrow

Encryption in this PKI allows for any pair of registered usersto communicate securely provided they share a common cryptographic engine. When key escrow is mandatory, each user will be required to provide a copy of his secret encryption key(s) to one or more certified Escrow Agents (EA). When not mandatory, a user may choose to escrow his keys with a trusted third party or even act as his own Escrow Agent (in such a case split escrow may not offer any advantage).

In the following all arithmetic is done modulo a universal prime number p . Registering for encryption services requires:

- User A generates two secret numbers u_A and v_A
- User A escrows his secrets with the split Escrow Authorities E_1 and E_2 by generating two additional secret numbers u_1 and u_2 (which he must securely store for possible data recovery). Each Escrow Agent will have published a universal parameter g^{r_1} and g^{r_2} respectively, which allows User A to form the values $g^{r_1 u}$ and $g^{r_2 u_2}$. These values will be used to form keys to encrypt his secret information for sending to the Escrow Agents.
- User A sends tto his Escrow Agents:

$$A \to E_1 \colon \langle g^{u_1}, E^{u_1 r_1}(u_A), ID_{CA},...\rangle_A$$

$$A \to E_2 \colon \langle g^{u_2}, E^{u_2 r_2}(v_A), ID_{CA},...\rangle_A$$

  where E is a known, and fixed, encryption algorithm used by the Escrow Agents. The session key for the encryption is $g^{u_i r_i}$.
- Upon receipt, each Escrow Agent decrypts its part of the split secret, and computes g^{u_A} and g^{v_A}, respectively. Each archives the information sent by user A and sends back to User

A's CA the package

$$\langle g^{u_A}, ID_A\rangle_{E_1}$$

$$\langle g^{v_A}, ID_A\rangle_{E_2}$$

respectively.

- Upon receipt of this information the CA is ready to register the user. It computes and forms the signed public key certificate

$$\langle\langle g^{u_A}, ID_A\rangle_{E_1}, \langle g^{v_A}, ID_A\rangle_{E_2}, ID_A, \ldots\rangle_{CA}$$

which it transmits to both User A (for verification) and to the Network Directory Server for placement in the public directory.

NOTES: A unique feature of this key escrow mechanism is that users must escrow their keys to get valid certificates. There is no way for the Certificate Authority to unilaterally enroll a user.

2. Encryption

Let us assume User B wishes to send User A a secure message. The following sequence is then followed:

- User B queries a Network Directory Server for User A's public key certificate.
- Upon receipt of User A's certificate User B computes a session key by forming

$$SK = H\left\{H[H(g^{u_A u_B}, ID_B, ID_A, \text{month}),\text{day}] \oplus H[H(g^{v_A v_B}, ID_B, ID_A, \text{month}),\text{day}],\text{random}\right\}$$

where H is a commonly held hash function, the month field is eleven bits (allowing for 2048 possibilities in this proposal, but may be any length), the day is five bits, and the random field is at least forty bits (to avoid the same session key if many messages are sent by User B to User A on the same day).

- User B encrypts the signed message under the session key SK and sends to User A

$$\text{control bytes,month,day,random,}E^{SK}(\langle m\rangle_B)$$

where the *control bytes* indicate what encryption was used, the key lengths, and how to process the succeeding bytes. (User B might also send his public key certificate to avoid a network lookup by User A.) Alternatively, he may choose to use SK as a *key encryption* *key* (KEK) instead. In this case the message is encrypted with a random key, and an encrypted version of this random key is sent with the message. This latter encryption is performed under the session key.

- Upon receipt, User A computes the session key by forming $(g^{u_B})^{u_A}$ and $(g^{v_B})^{v_A}$. User A next decrypts the message and then authenticates it by checking that the signature is valid.

NOTES: The formula for session key is new. It allows for long term caching of the most computationally intensive part of the key. Thus, users who communicate often do not have to recompute the entire session key. The structure of the session key also allows for time bounded warrents for decryption. The only other scheme along these lines (Yacbi, Lenstra, Winkler) requires Users A and B to each send the other specific data, after which, on the third communication, the actual message is transmitted. We avoid these two exchanges.

3. Law Enforcement

The specific form of the session key allows for law enforcement to read traffic without explicitly being given either pair u_A, u_B or v_A, v_B. In particular, suppose a warrant is issued to read all the traffic sent by User B to User A during the month of March 1996. In terms of User A's Escrow Agents, the following sequence is then followed:

- User A's split Escrow Agents E_1 and E_2 are served the warrant.
- E_1 computes its half of the split key escrow u_A and sends to law enforcement (signed and encrypted):

$$E_1 \rightarrow \text{Law Enforcement: } \langle H(g^{u_A u_B}, ID_B, ID_A, \text{month}), ID_B, ID_A \rangle_{E_1}$$

- Similarly E_2 computes its half of the split key escrow v_A sends to law enforcement

$$E_2 \rightarrow \text{Law Enforcement: } \langle H(g^{v_A v_B}, ID_B, ID_A, \text{month}), ID_B, ID_A \rangle_{E_2}$$

- Each day law enforcement hashes the day field into each of these packets, adds them together, and hashes in the per message random field to generate a session key.

At this point law enforcement can read only that traffic specifically covered by the warrant. *No secret keys are revealed.* If the warrant covers a specific set of days, or even an extended period of time, the Escrow Agent generates several key packets that cover the specified time period. Similarly, key packets will have to be generated for each user and for each direction of transmission covered by the warrant. [redacted] Such circumstances should be extremely rare and governed by very stringent requirements.

(b)(1)
OGA

FBI

Once the warrant expires, the user has no need to generate new [redacted]

(b)(1)
OGA

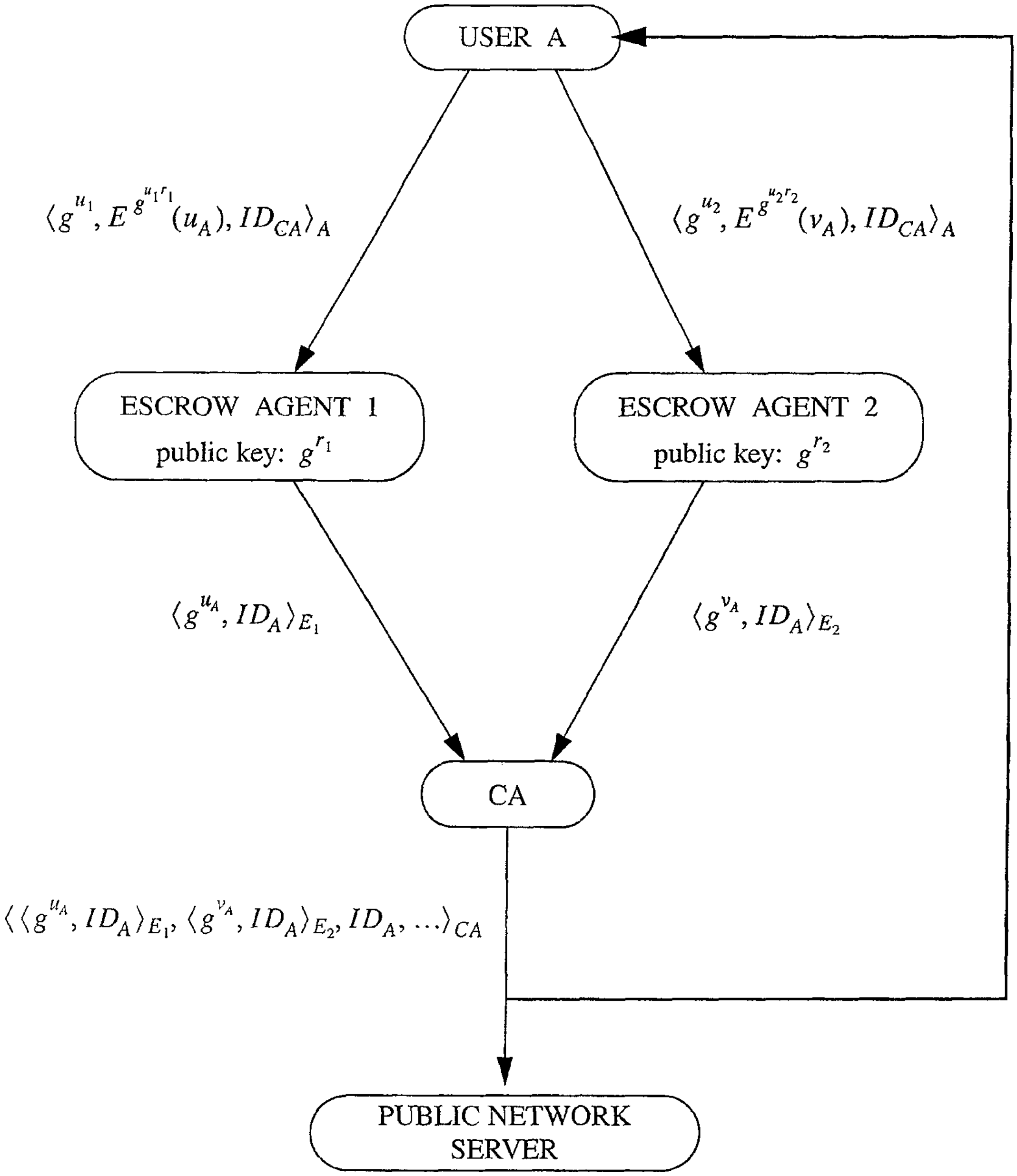
FBI

[redacted]

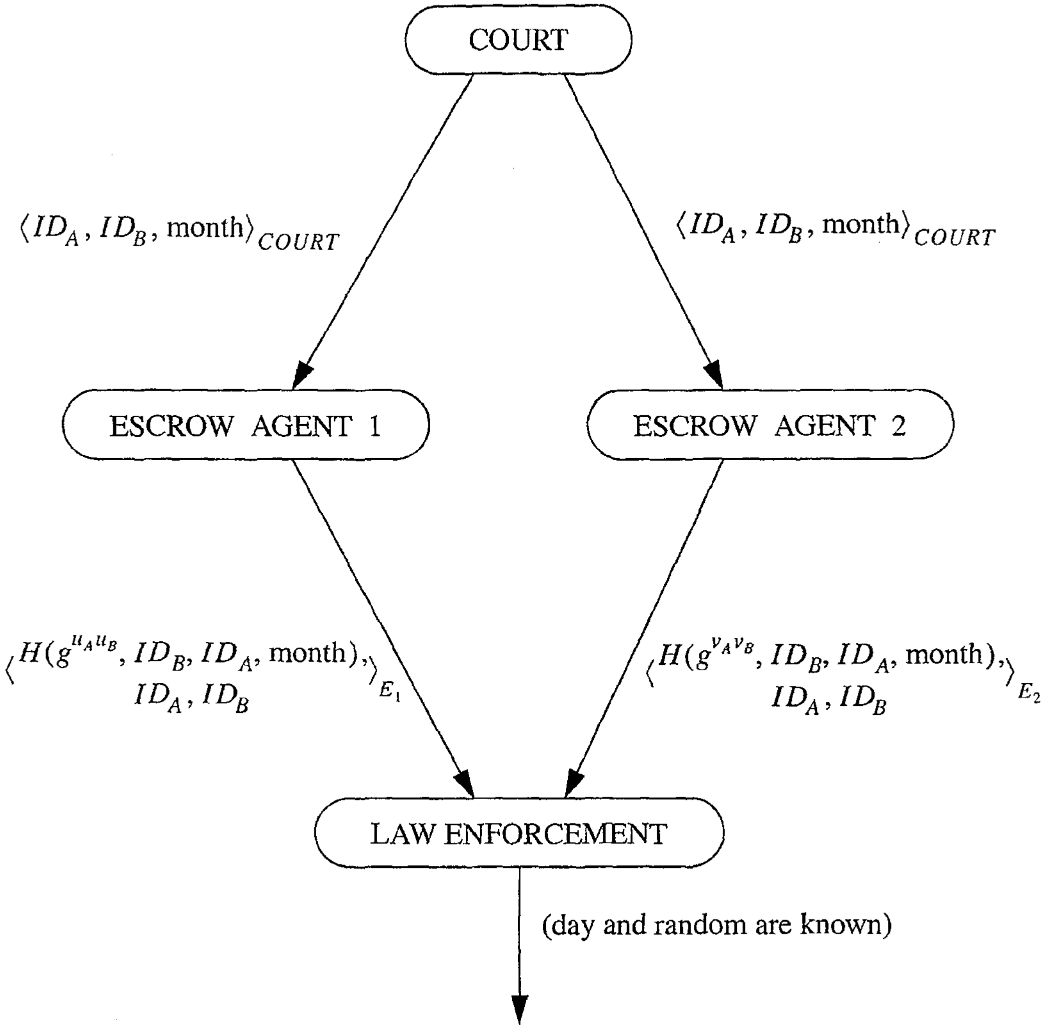
4. Data Recovery

Users may contract with a Data Recovery Center (DRC) to provide data recovery services for their data at rest. The DRC will be a valid user in the PKI, complete with a public key certificate(s) and public signature certificate(s). To save an encrypted file, the user will encrypt it with the session key derived from the DRC's public key. In other words, the session key will be derived *as if* the data was being encrypted *for* the DRC--only it will not be sent to the DRC. If data recovery is needed, the user will forward the file header information to his DRC. The DRC, in turn, will compute from the header the common session key and return this value to the user. With the session key, the user can now read his file.

NOTES: Data recovery is distinct from key escrow - a unique feature of this PKI. Also, the DRC does not need to decrypt any messages for the user, it merely gives the user the session key used to encrypt the file. The user, then, decrypts his own file--another unique feature. However, given access to the cipher, the DRC can decrypt the file.


USER A
$\langle g^{u_1}, E^{g^{u_1 r_1}}(u_A), ID_{CA}\rangle_A$
$\langle g^{u_2}, E^{g^{u_2 r_2}}(v_A), ID_{CA}\rangle_A$
ESCROW AGENT 1
public key: g^{r_1}
ESCROW AGENT 2
public key: g^{r_2}
$\langle g^{u_A}, ID_A\rangle_{E_1}$
$\langle g^{v_A}, ID_A\rangle_{E_2}$
CA
$\langle\langle g^{u_A}, ID_A\rangle_{E_1}, \langle g^{v_A}, ID_A\rangle_{E_2}, ID_A, \ldots\rangle_{CA}$
PUBLIC NETWORK SERVER

Key Escrow Mechanism


COURT
$\langle ID_A, ID_B, \text{month} \rangle_{COURT}$
$\langle ID_A, ID_B, \text{month} \rangle_{COURT}$
ESCROW AGENT 1
ESCROW AGENT 2
$\langle H(g^{u_A u_B}, ID_B, ID_A, \text{month}), ID_A, ID_B \rangle_{E_1}$
$\langle H(g^{v_A v_B}, ID_B, ID_A, \text{month}), ID_A, ID_B \rangle_{E_2}$
LAW ENFORCEMENT
(day and random are known)

$$SK = \begin{array}{l} H(\, H(H(g^{u_A u_B}, ID_B, ID_A, \text{month}), \text{day}) \oplus \\ \quad H(H(g^{v_A v_B}, ID_B, ID_A, \text{month}), \text{day}), \text{random}\,) \end{array}$$

Law Enforcement Access

A Modular Public Key Infrastructure
for
Security Management

A Modular Public Key Infrastructure for Security Management

Abstract

A modular public key infrastructure is outlined which supports key escrow for law enforcement, a separate data recovery component for restoration of archived information, secure message encryption, and strong authentication. The design is flexible in concept and is based on publicly established cryptographic algorithms. One strength of the system is that certifying authorities and data recovery centers never have access to users' secret keys, nor are they revealed in the warrant process. Moreover, warranted law enforcement access is limited both in time and direction by the public key mechanism.

1. Introduction

Any design of a public key infrastructure (PKI) must delicately balance the needs of industry, government, law enforcement, and the individual user community. The evolution of fast, reliable networks and the increasing reliance of society on them to conduct business present unique challenges to industry and government to guarantee authenticated, secure communications while providing legitimate access for warranted law enforcement. Moreover, this must be accomplished on a global scale.

As the electronic community evolves, so must the infrastructure that supports it. As such, the PKI described herein is *modular* in form allowing for components to be mixed and matched as appropriate. Thus, Escrow Agents serve only to process law enforcement warrants (and perhaps to restore lost secret encryption keys to users), Certificate Authorities serve only to authenticate users, and Data Recovery Centers serve only to restore archived information. Any subset of these components may be deleted from the PKI while still preserving its functionality (albeit with fewer services). Moreover, it is *scalable* both on the protocol level (key sizes, bit fields, etc.) and the user level (multiple Escrow Agents, Certificate Authorities, etc.).

In the following we describe each of the PKI components as they would interact in the context of a larger security management infrastructure. We do not prescribe any particular encryption algorithms, signature schemes, or hash functions in this proposal as these will no doubt need to evolve through consultation between industry, government, and the user community. However, it is important that a standard set of protocols and algorithms be defined—we only require that, upon their definition, they be implemented amongst the suite of options included in any PKI compliant product.

The design of this PKI was undertaken assuming a number of criteria had to be met. Nevertheless, this still provided for many choices and, when presented with such, a decision was generally made in favor of decreased network and computational overhead, user friendliness, and interoperability. Flexibility was a design requirement whenever possible so that, for example, the needs of law enforcement, the courts, and users are robustly drawn in the warrant process: only those communications precisely specified by the warrant can be deciphered by any parties other than the communicants.

Criteria

The following is a list of general design criteria:

- All components of the PKI will be made public.
- A key escrow mechanism, in which participants agree to escrow their secret encryption key(s) for warranted law enforcement, must be accommodated.
- The required default protocols and algorithms have been thoroughly vetted in the public domain.
- Encryption algorithms are essentially unregulated, but must be registered. Encryption packages must always include the default algorithms.

- Data recovery of encrypted files is accommodated by a mechanism distinct from the Escrow Authority (i.e., the Escrow Agents are not a part of the data recovery process).
- The infrastructure must be scalable to accommodate a large number of users.
- Secure communication must not *require* the recipient to play an active part in the key exchange (i.e., a common session key can be computed by the sender alone).
- Interoperability, at the cost of increased overhead, can be achieved with *key encryption key* (KEK) systems, such as *Royal Holloway.*

Features

Some of the design features of the PKI include:

- Signature keys are not escrowed in any way: private signature keys are never shared.
- None of the Data Recovery Centers, Certificate Authorities, warranted law enforcement entities, Escrow Agents, nor legitimate users of the PKI can digitally sign as another user.
- Recovery of a user's *encryption* secrets within this design requires the (illegal) collusion of the users Escrow Agents together with his Certificate Authority.
- Isolation of Certificate Authorities from knowledge of users' encryption *and* signing secrets reduces their liabilities and overhead, promotes more public trust, and allows them to focus on authentication.
- Data recovery of encrypted files is accommodated by a mechanism distinct from the Escrow Authority.
- Law enforcement is symmetric, i.e., the communications between two users can be legally monitored via access to *either* user's Escrow Agents.
- The key escrow mechanism requires very little software overhead.
- Users have the option of generating their own private encryption secrets or having these secrets generated for them.
- Infrastructure overhead per message has been minimized (e.g., no LEAF).

(b)(1)
OGA

FBI

[redacted]

- A data recovery center (DRC) can (and most likely will) be distinct from an Escrow Agent. As a DRC, it need only protect its own secret encryption key. Thus, it avoids many of the legal responsibilities (and costs) associated with an Escrow Agent.
- Default protocols are specified throughout guaranteeing any two users a secure communication path.
- Certificate Authorities can supply signed public key encryption certificates only after escrow requirements have been satisfied.
- Dishonest users cannot bypass escrow in communications with legitimate users (unlike LEAF systems).

In order for a user level software package to be certified PKI compliant (and eligible for export) this proposal requires

- Software must be registered with Department of Commerce, together with documented source code and encryption executables.
- All default algorithms (for hashing, signing, and encryption) must be implemented.
- User level software must check that incoming protocols are as expected (date stamp is accurate, signatures and certificates are valid) and notify the user of anomalies.
- Software must enforce fully qualified user identities (user@xyz.com.us).

2. Authentication

As with other specific algorithms implemented under this PKI, we leave the choice of a secure protocol for signing messages (e.g., the federal *Digital Signature Algorithm*) to the collaboration of industry, government, and users' groups. The Certificate Authority (CA), whose roles and responsibilities are detailed below, must have a mechanism for identifying a particular user to the network infrastructure by binding a public signature key to the user's identity. Once this is accomplished, the user can send and receive authenticated cleartext messages across the network.

Authentication is truly the bedrock on which the PKI is founded. Thus, each user's secret signing key is his most important property, for without it he cannot prove who he is, and in the possession of an untrusted third party it allows that third party to masquerade as the individual. In this proposal, *users are entrusted with securing their own signing keys*. (This is not to preclude that a user might want to enlist the services of an Escrow Agent for this purpose — but it is not a requirement of the PKI).

A user follows the following procedure to enroll in the PKI:

- The user presents identification and public signing parameters to a certified CA (This might have to be done physically, with a floppy disk or smart card, or by a notarized document exchange through the mail. A Policy Authority will have to set these guidelines.).
- The CA forms a public signature certificate binding the user's identity to his public key:

$$\langle ID_A, \text{public signature key, expiration date, ...} \rangle_{CA}$$

  where $\langle\ \rangle_{CA}$ means the contents are signed by the CA, and ID_A is user A's identity. (The precise format of the certificate may, for example, conform to the X.509 protocol.)
- The CA sends the certificate to a Network Directory Server (or several such servers) which are designated *authoritative* for that user, or CA.
- Upon compromise, revocation, change, or expiration of a signature certificate, the *CA* is responsible for immediately notifying the user's authoritative servers of a change in status.

The Network Directory Servers (NDS) may be configured to operate like DNS (*Domain Name Service*) so that together they form a distributed data base of user signature certificates. In this scenario an individual NDS services requests for which it is authoritative and may also choose to cache frequently requested certificates for which it is not authoritative.

In order to handle the case that a cached certificate may have been revoked (analogous to the situation in which a purchase is attempted with a credit card that has recently been stolen or has expired) a request for an authoritative response (i.e., directly to the user's authoritative NDS) should be supported. Moreover, the Policy Authority should establish *time to live* guidelines to prevent the long term caching of certificates. To prevent a malicious replay of an authoritative response (within the time to live period) a CA may choose to offer a notary service that provides a sender or receiver with a time stamped, signed, public key certificate for any user in its domain.

The implicit tree structure from the root Policy Authority down through the Certification Authorities to users enables a chain of trust to be established between any two users. For example, if User A has trusted Certification Authority CA_A while User B has trusted Certification Authority CA_B (all under the Policy Authority PA), then there exists a common parent CA (which may be PA itself) to both CA_A and CA_B. Trust of a parent of a trusted Certification Authority is implicit since the binding of an identity to a public signature key is based on the signature of the parent.

This chain of trust can be extended between different domains (i.e., between users under different Policy Authorities) if PAs cross-certify each other. This requires mutual agreement between domains (e.g., between different countries) that in particular includes agreement on signing parameters. Of course, cross-certification may take place at the Certification Authority level as well, say between CAs of a given company either within the same domain or between subsidiaries in different geopolitical domains.

3. Key Escrow

Encryption in this PKI allows for any registered user to communicate securely with any other registered user provided they share a common cryptographic engine. When key escrow is mandatory, each user will be required to provide a copy of his secret encryption key(s) to one or more certified Escrow Agents (EA). When not mandatory, a user may choose to escrow his keys with a trusted third party or even act as his own Escrow Agent (in such a case split escrow may not offer any advantage). The following escrow mechanism is independent of this choice.

The following description assumes *split key escrow*, i.e., a user constructs two secrets and shares one secret with each of two Escrow Agents. There is no design requirement for split escrow. Indeed, keys can be split between one, two, or many Escrow Agents and users with differing numbers of escrow agents can be easily accommodated. However, there are clear benefits to be had with two and that is the system advocated here. In addition, it is understood that all arithmetic is computed modulo a fixed, universal prime p together with a universal base g. The sizes and structure of these universal parameters will have to obey well established, fully vetted mathematical guidelines and be constructed in a manner that is acceptable to the PKI community. For example, the use of an elliptic curve group for exponentiation (in this case g is a base point for a universally known elliptic curve and exponentiation is usually written as multiplicaton) may offer valuable overhead savings with no attendant loss of security when compared to exponentiation in the ring of integers modulo p.

We follow the standard convention that when a quantity is framed by <> it is presumed to be digitally signed.

To register for encryption services:

- User A generates two secret numbers u_A and v_A which form the basis for all his encryption services.
- User A escrows his secrets with the split Escrow Authorities EA_1 and EA_2 by generating two additional secret numbers u_1 and u_2 (which he must securely store for possible data recovery). Each Escrow Agent will have published a universal parameter g^{r_1} and g^{r_2} respectively, which allows User A to form the values $g^{r_1 u}$ and $g^{r_2 u_2}$. These values will be used to form keys to encrypt his secret information for sending to the Escrow Agents.
- User A sends the following package to his Escrow Agents:

$$A \to EA_1 \colon \langle g^{u_1}, E^{g^{u_1 r_1}}(u_A), ID_{CA}, \ldots \rangle_A$$

$$A \to EA_2 \colon \langle g^{u_2}, E^{g^{u_2 r_2}}(v_A), ID_{CA}, \ldots \rangle_A$$

  where E is a known, and fixed, encryption algorithm used by the Escrow Agents. The session key for the encryption is $g^{u_i r_i}$.
- Upon receipt, each Escrow Agent decrypts its part of the split secret, and computes g^{u_A} and g^{v_A}, respectively. Each archives the information sent by user A and sends back to User A (or his CA) the package

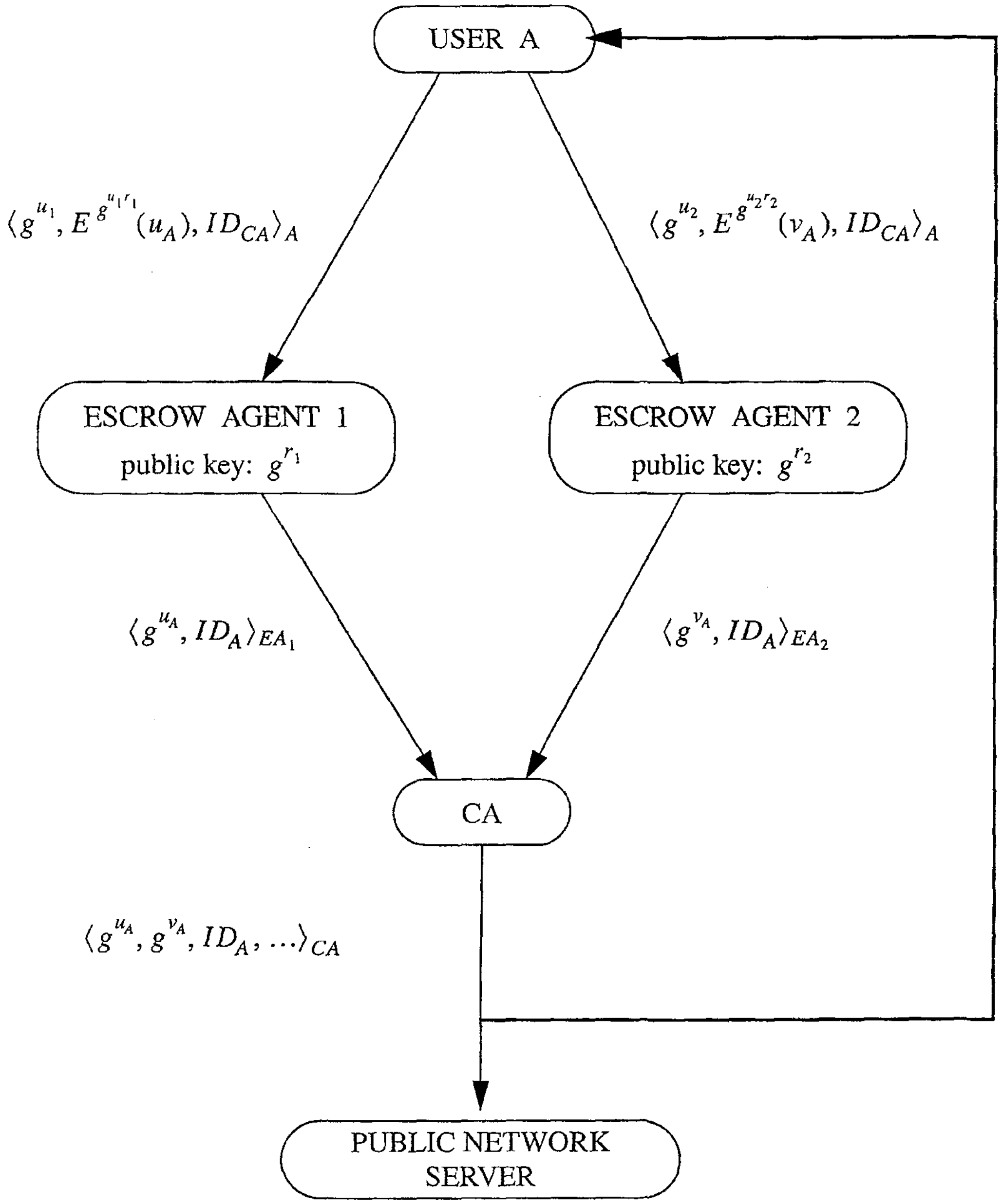
$$\langle g^{u_A}, ID_A \rangle_{EA_1}$$

$$\langle g^{v_A}, ID_A \rangle_{EA_2}$$

  respectively.
- Upon receipt of this information, either by User A or by User A's Escrow Agents, the CA computes and forms the signed public key certificate

$$\langle g^{u_A}, g^{v_A}, ID_A, \ldots \rangle_{CA}$$

  which it transmits to both User A (for verification) and to the Network Directory Server for placement in the public directory. The CA also records the identities of User A's Escrow Agents.


USER A
$\langle g^{u_1}, E^{g^{u_1 r_1}}(u_A), ID_{CA} \rangle_A$
$\langle g^{u_2}, E^{g^{u_2 r_2}}(v_A), ID_{CA} \rangle_A$
ESCROW AGENT 1
public key: g^{r_1}
ESCROW AGENT 2
public key: g^{r_2}
$\langle g^{u_A}, ID_A \rangle_{EA_1}$
$\langle g^{v_A}, ID_A \rangle_{EA_2}$
CA
$\langle g^{u_A}, g^{v_A}, ID_A, \ldots \rangle_{CA}$
PUBLIC NETWORK SERVER

Key Escrow Mechanism

4. Encryption

The encryption module is intended to be independent of whether or not there is mandatory key escrow. If escrow is not mandatory, user A will generate his secrets as described above but will bypass the Escrow Agent mechanism and directly send his (authenticated) public encryption keys to his CA. In either case, the CA registers user A's signed public encryption key certificate with an authoritative Network Directory Server(s).

The PKI is designed to allow any two users to construct a common *session key* (SK) for encryption. The SK can be used directly to key their common encryption algorithm or it can be used as a *key encryption key* (KEK). In the latter scenario, the sender generates a random encryption key and uses it to encrypt his message. He then sends $E^{SK}(\text{encryption key})$ along with the message. Using the common SK, the recipient decrypts this last packet, giving him the encryption key and therefore the ability to decrypt the message. On the surface, this adds unnecessary overhead to both the users and the network. However, there are advantages to using a KEK, including interoperability with other PKI proposals and reduced overhead for the user who wants to send the same message to many individuals.

Once User A and User B have been issued public key certificates (and have a common encryption package which may be the default standard, e.g., DES) they are ready to communicate securely. Let us assume User B wishes to send User A a secure message. The following sequence is then followed:

- User B queries a Network Directory Server for User A's public key certificate.
- Upon receipt of User A's certificate User B computes a session key by forming

$$SK = H\left\{H[H(g^{u_A u_B}, ID_B, ID_A, \text{month}),\text{day}] \oplus H[H(g^{v_A v_B}, ID_B, ID_A, \text{month}),\text{day}],\text{random}\right\}$$

where H is a commonly held hash function, the month field is eleven bits (allowing for 2048 possibilities in this proposal, but may be any length), the day is five bits, and the random field is at least forty bits (to avoid the same session key if many messages are sent by User B to User A on the same day).

- User B encrypts the signed message under the session key SK and sends to User A

$$\text{control bytes,month,day,random,}E^{SK}(\langle m\rangle_B)$$

where the *control bytes* indicate what encryption was used, the key lengths, and how to process the succeeding bytes. (User B might also send his public key certificate to avoid a network lookup by User A.) Alternatively, he may choose to use SK as a *key encryption key* (KEK) instead. In this case the message is encrypted with a random key, and an encrypted version of this random key is sent with the message. This latter encryption is performed under the session key.

- Upon receipt, User A computes the session key by forming $(g^{u_B})^{u_A}$ and $(g^{v_B})^{v_A}$. User A next decrypts the message and then authenticates it by checking that the signature is valid.

5. Law Enforcement

This PKI is designed to provide maximum flexibility for the needs of both law enforcement and the courts provided key escrow is enforced. In particular, it offers many layers of granularity in providing law enforcement warranted access to encrypted communications and/or encrypted archives, [redacted]

$$SK = H\left\{ H[H(g^{u_A u_B}, ID_B, ID_A, \text{month}),\text{day}] \oplus H[H(g^{v_A v_B}, ID_B, ID_A, \text{month}),\text{day}],\text{random} \right\}$$

(b)(1)
OGA

where

- SK is the session key (or key encryption key) used to encrypt the particular message,
- H is a one-way hash function (e.g., SHA, the Federal standard Secure Hash Algorithm)

FBI

[redacted]

- The month field is comprised of eleven bits which identifies up to 2048 months both past and present, the day field is five bits, and the random field is at least forty bits.

The session key is *asymmetric* in that the order of ID_A and ID_B indicates the direction of the transmission. Thus, bidirectional traffic between two users on the same day that uses the same forty bit random pattern will use different session keys. However, since g^{u_A}, g^{v_A} and g^{u_B}, g^{v_B} are publicly known, knowledge of either pair u_A, v_A or u_B, v_B is sufficient to produce a session key given the month, day, and random fields. Therefore, a warrant served on either User A's or User B's Escrow Agent(s) will suffice to read their traffic.

The specific form of the session key allows for law enforcement to read traffic without explicitly being given either pair u_A, v_A or u_B, v_B. In particular, suppose a warrant is issued to read all the traffic sent by User B to User A during the month of March 1996. In terms of User A's Escrow Agents, the following sequence is then followed:

- User A's CA is queried for the identity of his Escrow Agents (the identity of the CA is manifest in User A's public key certificate).
- User A's split Escrow Agents EA_1 and EA_2 are served the warrant.
- EA_1 computes its half of the split key escrow u_A and sends to law enforcement (signed and encrypted):

$$EA_1 \rightarrow \text{Law Enforcement: } \langle H(g^{u_A u_B}, ID_B, ID_A, \text{month}), ID_B, ID_A \rangle_{EA_1}$$

- Similarly EA_2 computes its half of the split key escrow v_A sends to law enforcement

$$EA_2 \rightarrow \text{Law Enforcement: } \langle H(g^{v_A v_B}, ID_B, ID_A, \text{month}), ID_B, ID_A \rangle_{EA_2}$$

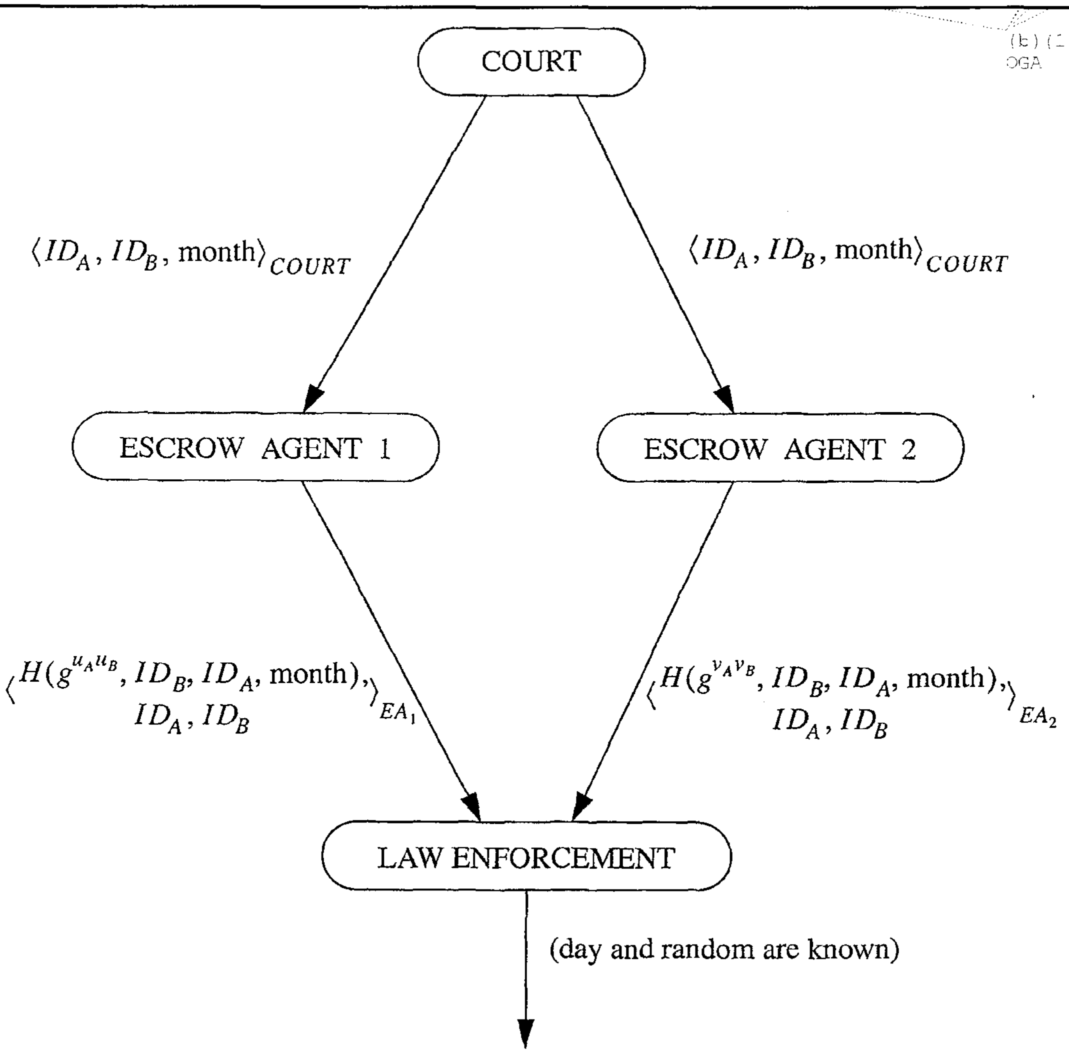
- Each day law enforcement hashes the day field into each of these packets, adds them together, and hashes in the per message random field to generate a session key.

At this point law enforcement can read only that traffic specifically covered by the warrant. *No secret keys are revealed.* If the warrant covers a specific set of days, or even an extended period of time, the Escrow Agent generates several key packets that cover the specified time period. Similarly, key packets will have to be generated for each user and for each direction of transmission covered by the warrant [redacted] [redacted] Such circumstances should be extremely rare and governed by very stringent requirements.

Once the warrant expires, the user has no need to generate new [redacted]

(b) (1)
OGA

FBI


COURT
$\langle ID_A, ID_B, \text{month}\rangle_{COURT}$
$\langle ID_A, ID_B, \text{month}\rangle_{COURT}$
ESCROW AGENT 1
ESCROW AGENT 2
$\langle H(g^{u_A u_B}, ID_B, ID_A, \text{month}), ID_A, ID_B\rangle_{EA_1}$
$\langle H(g^{v_A v_B}, ID_B, ID_A, \text{month}), ID_A, ID_B\rangle_{EA_2}$
LAW ENFORCEMENT
(day and random are known)

$$SK = H(\,H(H(g^{u_A u_B}, ID_B, ID_A, \text{month}), \text{day}) \oplus H(H(g^{v_A v_B}, ID_B, ID_A, \text{month}), \text{day}), \text{random}\,)$$

Law Enforcement Access

6. Data Recovery

In this PKI a user may contract with a data recovery center (DRC) to provide data recovery services for files encrypted on the user's disk. Outwardly, a DRC acts exactly like any user in the network: the DRC registers with a Certificate Authority, has its secret key(s) escrowed, and is a fully qualified user on the system. When a user wishes to store a file on his disk he encrypts it just as if he were sending it to his DRC, but writes it to his disk. Since the DRC is the legitimate target of the message, it can compute the session key that was used to encrypt it. Thus, if a user is unable to decrypt a disk file, he can send the file's header information (month, day, random) to his DRC who, in turn, can provide the user with the session key that decrypts the file.

From a law enforcement perspective, reading a user's disk files (given access to them) amounts to obtaining a warrant to read the traffic between the user and his DRC. Either the user's or the DRC's Escrow Agents can service this request. In this fashion, the DRC has no responsibility to provide services to law enforcement and must only protect its own [redacted]

FBI

According to this scenario, then, a user who receives a file encrypted with one session key will reencrypt the message with a new session key and write the result to disk. This latter session key can be reconstructed only by the user, his DRC, and the combined efforts of either's Escrow Agents when served with a warrant.

If a user does not elect to employ a DRC, he may simply archive received encrypted files, or in the case of encrypting a file for the first time do so as if mailing it to himself.

7. Roles and Requirements

Each of the component elements of the PKI has its own role and set of requirements in order for the infrastructure to interoperate efficiently. In this section we outline, in general terms, the major functions each component must provide.

Policy Authority (PA)

The Policy Authority establishes requirements for PKI components and certifies them compliant. Its role in the PKI is a continually evolving one that addresses and arbitrates the requirements of users, industry, government, and law enforcement. It is also responsible for:

- Fostering international agreements that promote interoperability.
- Developing and enforcing safeguarding standards for Escrow Agents.
- Determining default algorithms for digital signatures, encryption, and public key exchange.

- Establishing standards for Certificate Authorities and acting as their authentication agent.
- Developing new standards as the PKI infrastructure evolves.
- Issuing and responding to security alerts that affect the PKI.

Escrow Authorities (EA)

This PKI employs dual Escrow Agents within a given domain (domains will probably exist at the national level but may be distributed to lower echelons). The concept of split escrow immediately extends to any number of Escrow Agents servicing a particular user. Indeed, one can conceive of a single Escrow Agent (who now knows all of a user's secrets) to multiple Escrow Agents each archiving a proportional part of a user's secret encryption keys. In any scenario, the EA must provide a number of services:

- Escrow (and archive) users' (split) keys and protect them from unauthorized access.
- Process warrants for law enforcement.
- Provide Certificate Authorities with users' public key parameters which can then be signed and placed on the public network directory.
- Establish a secure channel to a user for exchanging data (the registration process requires that a user send his secrets to the EAs securely—the EA may have to provide publicly available software to accomplish this).
- Upon request, create secret encryption keys for a user.
- Be equipped to provide secret key recovery for each user in its domain (failure of a smart card, disk crash, etc. — this service may be at the Policy Authority's discretion).

Certificate Authorities (CA)

The role of a Certificate Authority in the PKI is made distinct from data recovery (although it may be a Data Recovery Center, as well) and encryption key escrow. It serves to authenticate users to the network by signing their public key certificates and forwarding their certificates to authoritative Network Directory Servers. It is conceivable that there may be multiple levels of authentication available corresponding to the level of rigor employed in identifying the user to the CA. Since only a CA is authorized to sign public key certificates, it must protect its signing secret(s) and assume some liability for their unauthorized disclosure. However, since it need protect no keys other than its own, the security requirements levied on a CA will be much less stringent than those of an Escrow Agent. Indeed, a CA might be little more than a secured workstation with appropriate network firewall protection. In practice, there will likely be a hierarchy of CAs that can establish a chain of trust between any two users. CAs will authenticate other CAs within their own domain with the top level CA being authenticated by a Policy Authority.

More specifically, a CA must provide the following services:

- Bind users' identities to their public signature keys (this may require physical identification — or more — for full authentication).

- Sign public key certificates (as received from the EAs in the case of encryption key escrow) and forward them to the user's authoritative Network Directory Server(s).
- Record the identities of each user's Escrow Agents, i.e., store $\langle g^{u_A}, ID_A \rangle_{EA_1}$ and $\langle g^{v_A}, ID_A \rangle_{EA_2}$. The availablity of this information outside of Law Enforcement will be determined by the Policy Authority.
- Escrow users' public key signature certificates for a time specified by law.
- Issue revocation certificates to the public network directory for users in its domain according to policies set forth by the PA.
- In accordance with PA policy, provide signed, time-stamped certificates of a user's signature key for revocation inquiries.

Public Network Service (PNS)

A Public Network Service maintains an up-to-date distributed directory of valid users together with their signed public signature and encryption certificates. This information is stored on Network Directory Servers in established formats and is accessed according to agreed to protocols (e.g., X.509). The PNS must also maintain key revocation lists for invalidated users and provide backup redundancy for network outages. The requirements for the PNS will be determined and enforced by the PA.

~~CONFIDENTIAL//MR~~

A Modular Public Key Infrastructure for Security Management

~~CONFIDENTIAL//MR~~

~~FOR OFFICIAL USE ONLY~~

Balancing Interests

NATIONAL SECURITY

- LAW ENFORCEMENT: Warranted access via escrow
- INFOSEC: Secure algorithms and protocols
- [redacted]

(b)(1)
(b)(3)-P.L. 86-36

INDUSTRY

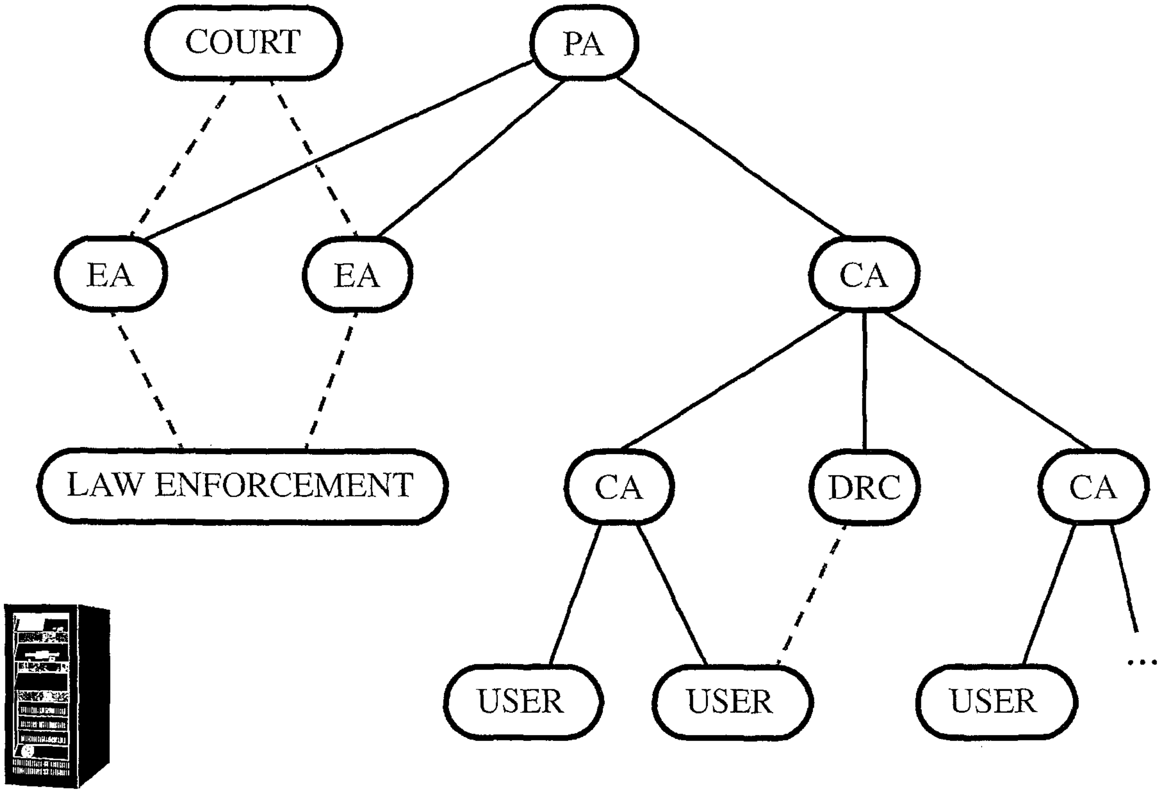
- Modular and flexible

USERS

- Strong authentication and privacy
- Data recovery

~~CONFIDENTIAL//MR~~

The Components

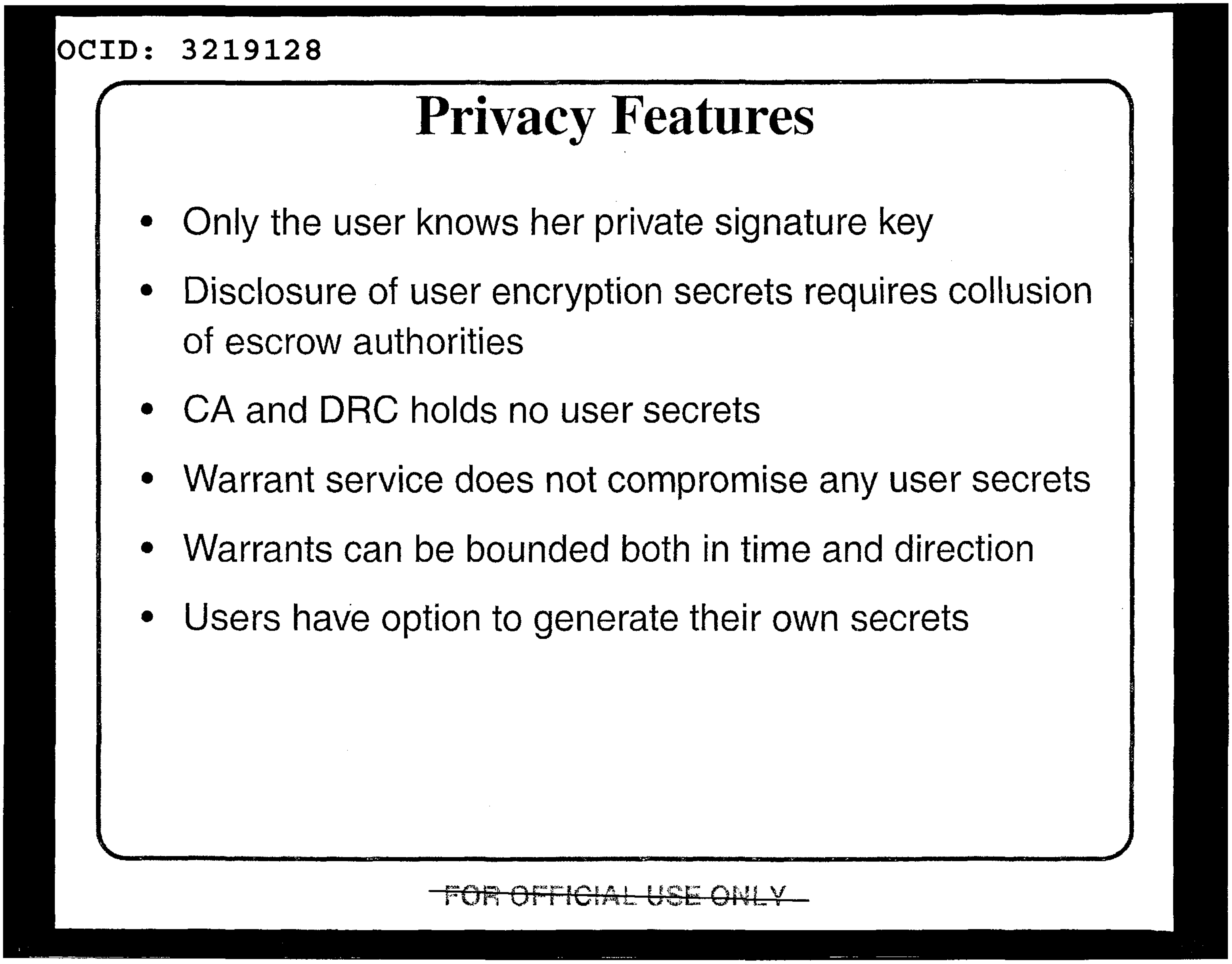


~~FOR OFFICIAL USE ONLY~~

Design Criteria

- Key-escrow capability
- All system components are public
- All default algorithms are well-known
- User has choice of encryption algorithm
- Data recovery distinct from escrow mechanism
- Encryption session key computed via one sided exchange (recipient does not have to *actively* participate)
- Scalable to large number of users
- Interoperable with other systems

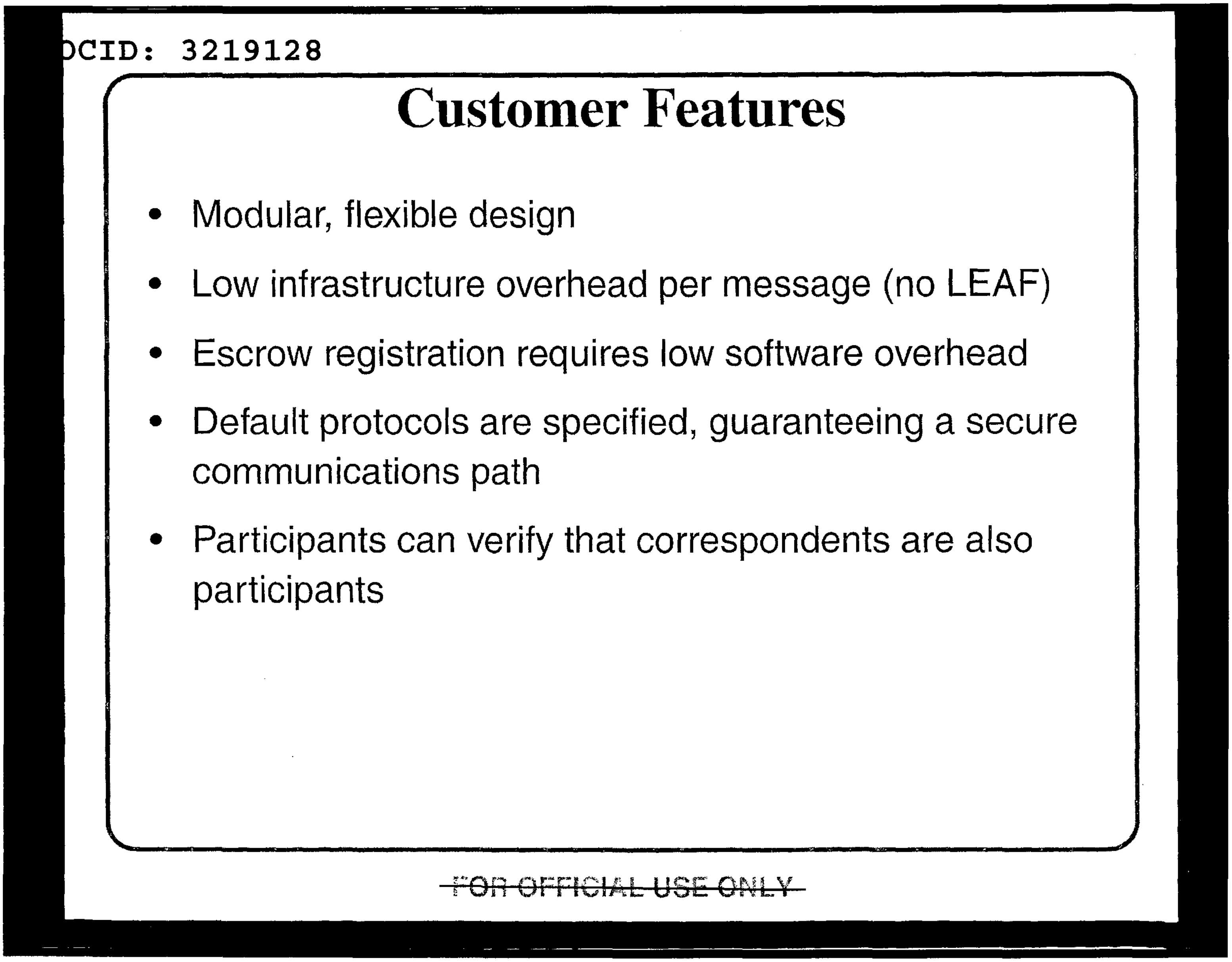
~~FOR OFFICIAL USE ONLY~~



Privacy Features

- Only the user knows her private signature key
- Disclosure of user encryption secrets requires collusion of escrow authorities
- CA and DRC holds no user secrets
- Warrant service does not compromise any user secrets
- Warrants can be bounded both in time and direction
- Users have option to generate their own secrets

~~FOR OFFICIAL USE ONLY~~



Customer Features

- Modular, flexible design
- Low infrastructure overhead per message (no LEAF)
- Escrow registration requires low software overhead
- Default protocols are specified, guaranteeing a secure communications path
- Participants can verify that correspondents are also participants

~~FOR OFFICIAL USE ONLY~~

Ingredients

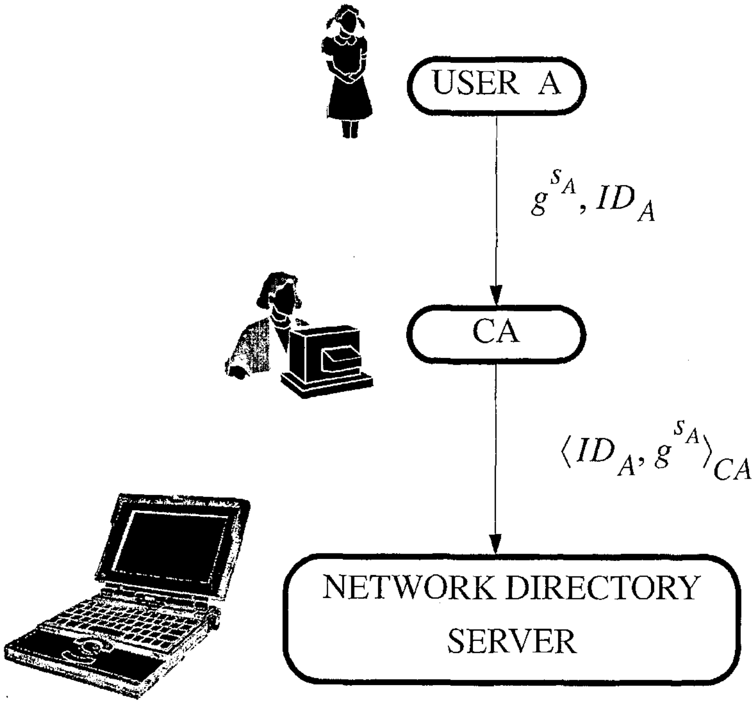
AUTHENTICATION

- Private and public signature key pair
- Proof of identity

ENCRYPTION

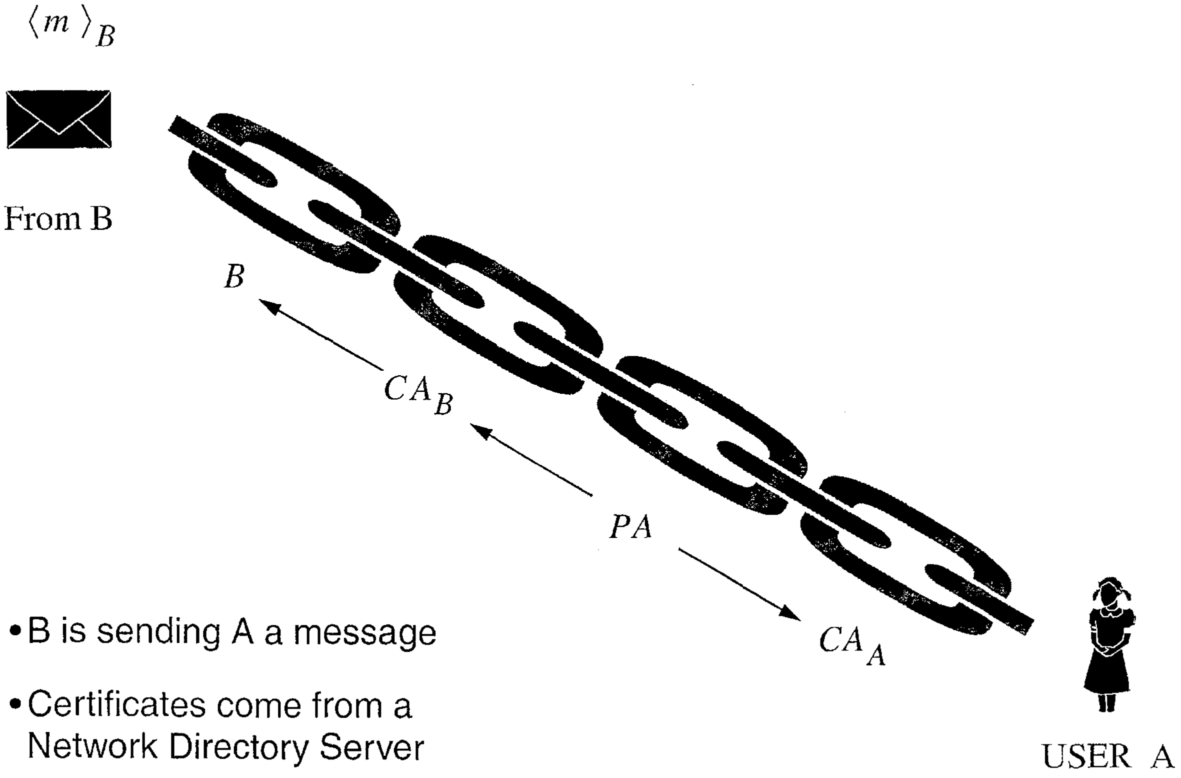
- Universal base g and prime p
- Escrowed, secret encryption keys u_A and v_A
- Public encryption keys g^{u_A} and g^{v_A}
- A secure hash algorithm H

Authentication: Enrollment



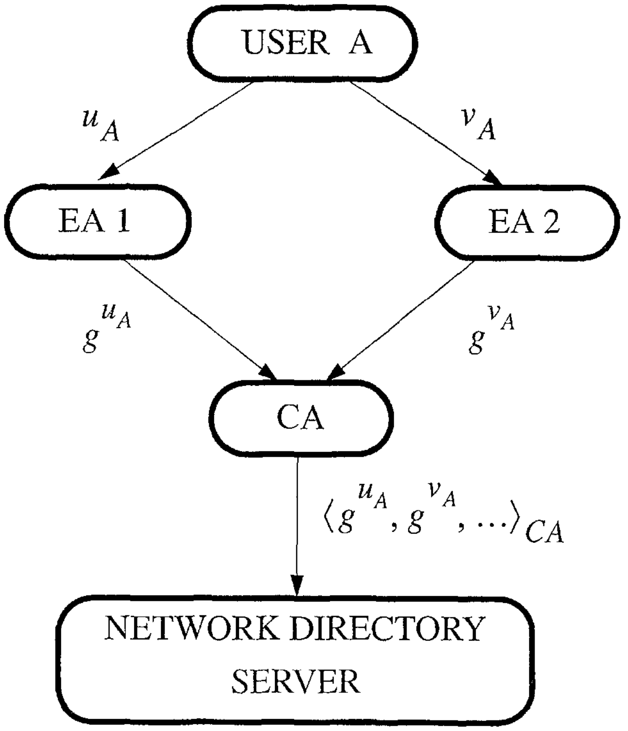
~~FOR OFFICIAL USE ONLY~~

Authentication: Chain of Trust



- B is sending A a message
- Certificates come from a Network Directory Server

Encryption Key Escrow/Enrollment



$\langle g^{u_1}, E^{g^{u_1 r_1}}(u_A), CA\rangle_A$

$\langle g^{u_2}, E^{g^{u_2 r_2}}(v_A), CA\rangle_A$

$\langle g^{u_A}, A\rangle_{EA_1}$

$\langle g^{v_A}, A\rangle_{EA_2}$

$\Big\langle\langle g^{u_A}, A\rangle_{EA_1},$

$\langle g^{v_A}, A\rangle EA_2, ..\Big\rangle_{CA}$

Encrypted Message from B to A

- Session Key:

$$SK = SK(B, A, \text{month}, \text{day}, \text{random})$$

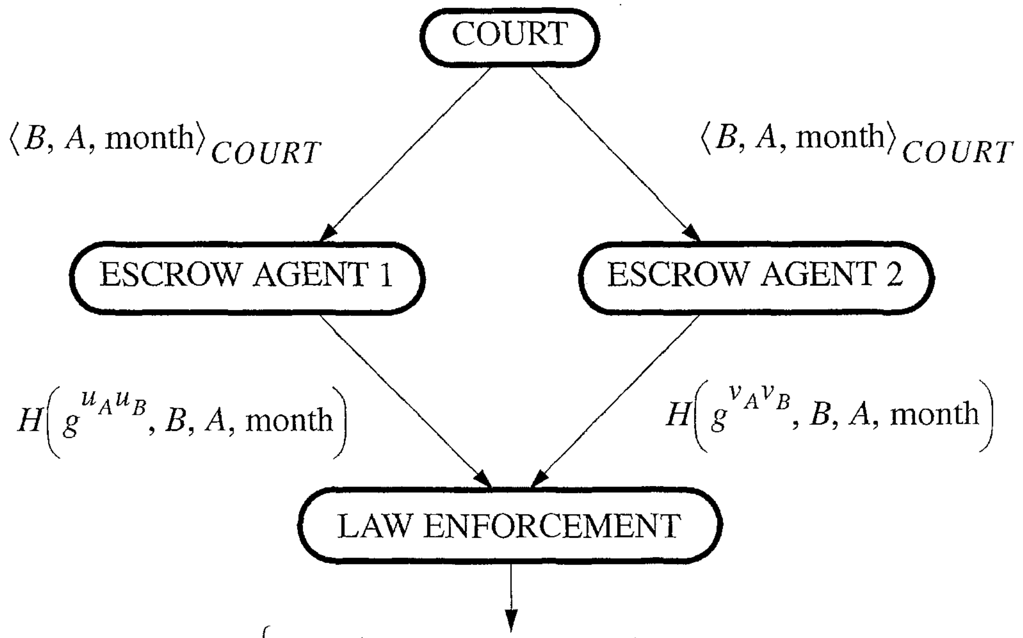
$$= H\left\{ H\left[H\left(g^{u_A u_B}, B, A, \text{month} \right), \text{day} \right] \oplus H\left[H\left(g^{v_A v_B}, B, A, \text{month} \right), \text{day} \right], \text{random} \right\}$$

- month, day, and random sent in clear:

$$\text{header, month, day, random, } E^{SK}(\langle m \rangle_B)$$

$$\text{header, month, day, random, } E^{SK}(CV), E^{CV}(\langle m \rangle_B)$$

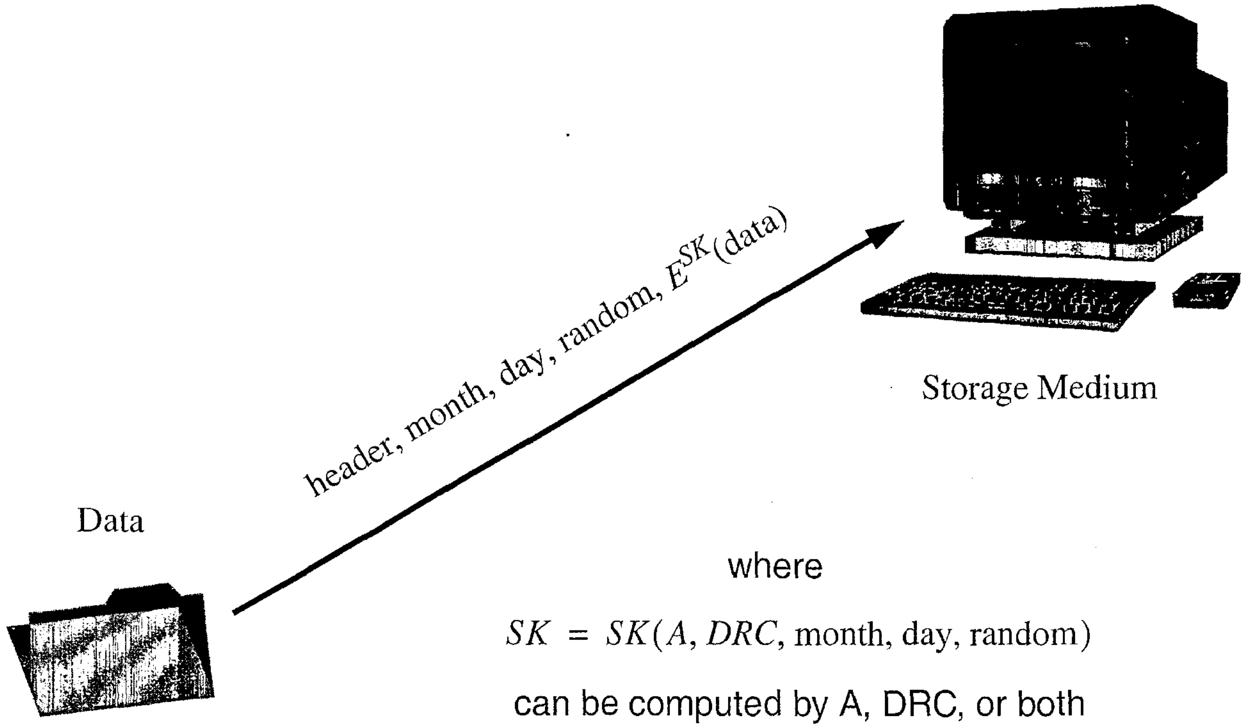
Law Enforcement Access



COURT
$\langle B, A, \text{month}\rangle_{COURT}$
$\langle B, A, \text{month}\rangle_{COURT}$
ESCROW AGENT 1
ESCROW AGENT 2
$H\left(g^{u_A u_B}, B, A, \text{month}\right)$
$H\left(g^{v_A v_B}, B, A, \text{month}\right)$
LAW ENFORCEMENT

$$SK = H\left\{ H\left[H\left(g^{u_A u_B}, B, A, \text{month}\right), \text{day}\right] \oplus H\left[H\left(g^{v_A v_B}, B, A, \text{month}\right), \text{day}\right], \text{random} \right\}$$

Data Encryption with Recovery



Data

header, month, day, random, E^{SK}(data)

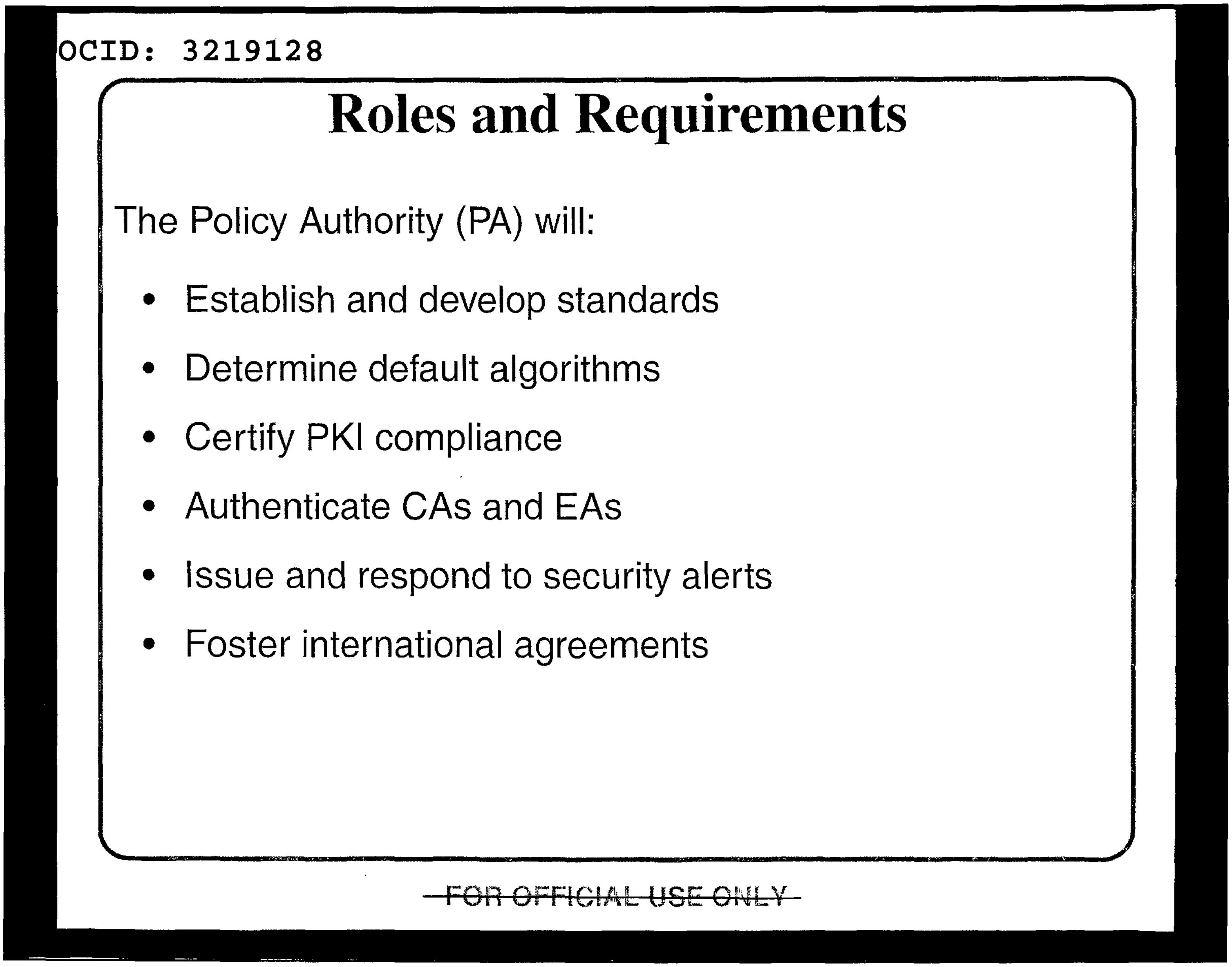
Storage Medium

where

$SK = SK(A, DRC, \text{month}, \text{day}, \text{random})$

can be computed by A, DRC, or both escrow agents of either, under warrant

~~FOR OFFICIAL USE ONLY~~

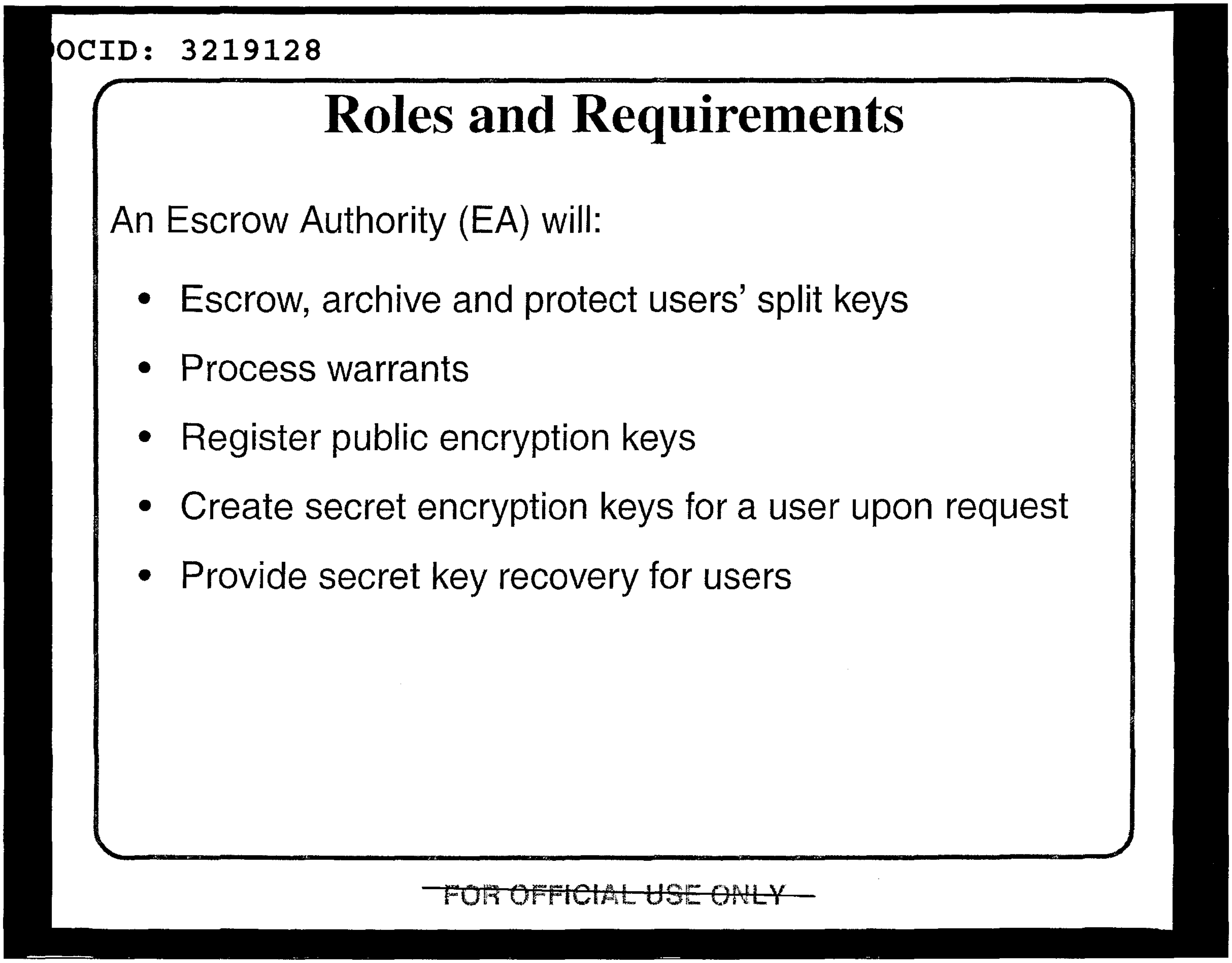


Roles and Requirements

The Policy Authority (PA) will:

- Establish and develop standards
- Determine default algorithms
- Certify PKI compliance
- Authenticate CAs and EAs
- Issue and respond to security alerts
- Foster international agreements

~~FOR OFFICIAL USE ONLY~~



Roles and Requirements

An Escrow Authority (EA) will:

- Escrow, archive and protect users' split keys
- Process warrants
- Register public encryption keys
- Create secret encryption keys for a user upon request
- Provide secret key recovery for users

~~FOR OFFICIAL USE ONLY~~

Roles and Requirements

A Certificate Authority (CA) will:

- Bind users' identities to public signature keys
- Sign public key certificates and post them to network
- Archive users' signature certificates for a time specified by law
- Issue revocation certificates to the public network